Selecting Youtube Video Thumbnails via Convolutional Neural Networks

Noah Arthurs*
Stanford University
narthurs@stanford.edu

Sawyer Birnbaum*
Stanford University
sawyerb@stanford.edu

Nate Gruver*
Stanford University
ngruver@stanford.edu

Abstract

The success of a Youtube channel is driven in large part by the quality of the thumbnails chosen to represent each video. In this paper, we describe a CNN architecture for fitting the thumbnail qualities of successful videos and from there selecting the best thumbnail from the frames of a video. Accuracy on par with a human benchmark was achieved on the classification task, and the ultimate thumbnail selector picked what we deemed "reasonable" frames about 80% of the time. In depth analysis of the classifier was also performed and data augmentation was used to attempt improvements on flaws noticed. Video category information was also incorporated into a later model in an attempt to create more semantically fitting thumbnails. Ultimately, the success of augmentation and additional semantic information at selecting good frames did not differ much from earlier results but revealed promising qualitative structures in the selection task.

1. Introduction

Every YouTube video is represented by a thumbnail, a small image that, along with the title and channel, serves as the "cover" of the video. Thumbnails that are interesting and well-framed attract viewers, while those that are confusing and low-quality encourage viewers to click elsewhere. As a testament to the important of a good thumbnail, 90% of the most successful YouTube videos have custom thumbnails [2]. YouTube uploaders without the time or skills to create a custom thumbnail, however, must pick one of 3 frames automatically chosen from the video. Our mission is to improve this frame selection process and help uploaders select high quality frames that will attract viewers to their channel.

We use a two phase process to select good thumbnails for a video. In the 1st phase, we train a convolutional neural network (CNN) to predict the quality of a video (encoded with a binary-good/bad label) from the video's thumbnail. In the next phase, we run a set of frames from a video through our model and use the softmax scores produced by the algorithm to rank their quality as thumbnails. We can then recommend the frames that achieve the highest scores.

*All authors contributed equally to this project

2. Related Work

Because of its importance to both content creators and hosts of video sharing platforms, thumbnail selection has become a major area of research in the last 10 years. Most early work in the field focused on thumbnail selection through more classical feature selection techniques [32] [15] [19]. The focus in these studies was primarily streamlining the thumbnail selection pipeline [9] [15] [31] [19] as well as selecting semantically relevant thumbnails through regression [32] or metrics of mutual information [20] [13]. Only recently have thumbnail selection systems begun to utilize convolutional neural networks. Here again we see three primary areas of focus. First, generation of thumbnails that are correlated with video success [28] [26], second, generation of thumbnails for semantic relevance using a mixture of CNNs and NLP [27] [21] [29] [30], and, third, thumbnail selection through measurable aesthetic qualities [24] [8].

In this work we decided to focus on the first of these areas, crafting thumbnails with an eye towards general video success. This makes our work most like that of Weilong Yang and Min-hsuan Tsai [28] at Google DeepMind. To build on their accomplishments, we preformed a thorough analysis of our model's inner working and employed data augmentation to help the model focus on thumbnail quality rather than on video branding (i.e., with logos). We also attempted to incorporate the semantic information encapsulated in the video's category.

3. Dataset and Methods

Our project consists of three parts:

1. Create a dataset of good and bad thumbnails.
2. Train a classifier to distinguish between these two categories.

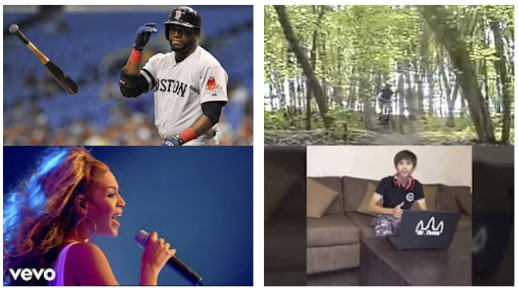
3. Select thumbnails by choosing from each video the frames that have the highest probability of being good thumbnails according to the classifier.

3.1. Dataset

For the 1st phase of our system, we gather thumbnails from "good" and "bad" videos. We defined "good" and "bad" as a function of the number of views received by a video, under the assumption that a video with a high view count likely has a custom, well-designed thumbnail (as mentioned above, this is true for 90% of the most popular videos), while videos with extremely few views likely have unappealing thumbnails. Specifically, we labeled thumbnails with more the 1 million views "good" and those with fewer than 100 views "bad."

To collect the good thumbnails, we downloaded (at most) 5 videos with a million or more views from the 2,500 most subscribed YouTube channels [4][1]. This method further ensures that the good thumbnails we selected are custom images designed by experienced YouTube content creators. To select bad thumbnails, we considered videos selected by a psuedorandom algorithm and included those with fewer than 100 views (which was about half of the total) [1].

This process provided us with $\sim$ 5000 videos of each class. In general, as expected, the good class thumbnails are noticeably higher-quality than the bad class ones, although there is a fair amount of noise in the data. We set aside 10% of the data for the test set and 20% of the data for the validation set.



Example thumbnails: good on the left and bad on the right

Both the good and bad thumbnails come from a diverse set of YouTube categories. The distribution of thumbnails, however, differs between the classes. Here are the most common categories for both classes (with associated frequency counts):

Frequency	Good	Bad
1	Music (1335)	Entertainment (2656)
2	Howto & Style (1048)	People & Blogs (807)
3	News & Politics (672)	Howto & Style (292)

[1]We downloaded YouTube video data (thumbnails, frames, etc.) using youtube-dl and the YouTube Developer API [6] [5]

While these distributional differences are a potential source of concern (we do not want the model to make label predictions based on predictions of thumbnails' category without considering their quality) the amount of diversity within each category and the similarities between some of the top categories, such as music and entertainment, force the model to evaluate images on more than a categorical level. In any event, our results suggest that the model is discriminating on more than just the image category. (See 4.2)

After downloading the thumbnails, we scale them to YouTube's recommended 16x9 resolution, cropping images if they are initially too tall and adding a black boarder if they are initially too wide. (This corresponds with how YouTube handles missized thumbnails.) The images are then resized to 80x45 pixels using Lanczos resampling [10] to reduce the size of the model and allow for efficient training. Image resizing was performed using SciPy [16]. Lastly, we zero-centered and normalized.

For the 2nd phase of our system, we downloaded 1 frame per second from 84 videos across the 9 most popular YouTube categories.[2]. We resized each frame to the same resolution as the thumbnail data and created a set of 10 frames per video. We spaced the frames out evenly for each video to capture a somewhat representative sample of the scenes in the video and help ensure that frames could be easily differentiated.

3.2. Classifier

Our classifiers are neural networks that take an image as input and output a two-dimensional vector of scores $S = (s_0, s_1)$, where s_0 is the score of the bad class and s_1 is the score of the good class. We turn these scores into "probabilities" using the softmax function. Specifically, the probability that the image is good (according to our model) is:

$$P(y = 1) = \frac{e^{s_1}}{e^{s_0} + e^{s_1}}$$

where y refers to the true class of the image in question. Optimally, then, the model should assign a score of $-\infty$ to the incorrect class and score of ∞ for the correct class for each example. In order to measure accuracy on the classification task, we say that our model is classifying an image as good if $P(y = 1) > 0.5$ and bad otherwise. Two of our group members tested ourselves on the classification task across 212 examples, and both achieved an accuracy of 81.6%, which we consider our human benchmark.

Our loss for example i of class y_i is given by the softmax cross entropy loss [12]:

$$L_i = -\log(P(y = y_i)) = -\log\left(\frac{e^{s_{y_i}}}{e^{s_0} + e^{s_1}}\right)$$

[2]Music, Comedy, Film & Entertainment, Gaming, Beauty & Fashion, Sports, Tech, Cooking & Health, and News & Politics

Our loss for a batch of n data points consists of the average cross entropy loss for that batch plus an $L2$ regularization term to encourage sparsity in the model and avoid overfitting [22]:

$$L = \frac{1}{n}\sum_{i=1}^{n} L_i + \lambda * \sum_{w} w^2$$

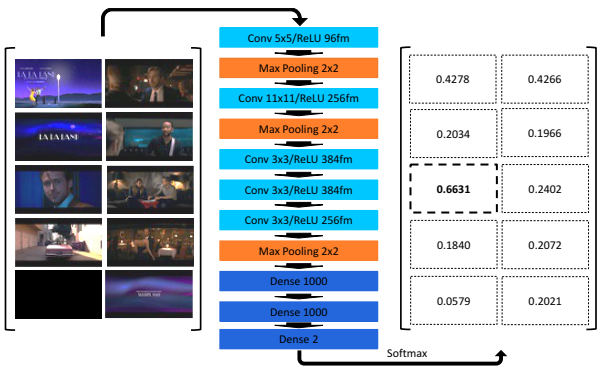
where λ is a regularization constant and the w's are the weights (not the biases) for the dense and convolutional layers. We used TensorFlow [7] to implement our models and an Adam Optimizer [17] to minimize the loss function so that we had an adaptive learning rate for each parameter.

Each network starts with a series of convolutional layers which downsample either by placing 2x2 max pooling layers between the convolutions or by performing convolution with a stride of 2. After the last convolutional layer, the activations are flattened and then put through a series of dense layers, the last of which produces a two-dimensional output, which are the scores.

In addition to $L2$ regularization, we used dropout [25] to prevent overfitting in our network. We were able to train effectively without Batch Normalization [14], so we did not include it.

3.3. Frame Selection

The pipeline for frame selection is depicted below. Frames are run through the pretrained model and the softmax score associated with each frame represents the quality confidence for each frame. The frames with the highest confidence are the ones judged most likely to be good by the model and are thus the ones picked for the video.



For the actual selection task, only 10 frames chosen linearly from the 100 extracted were utilized so that the results of the pipeline could be benchmarked against a human standard. To create the human benchmark, we collaboratively selected those frames out the 10 that were "reasonable" choices for the video's thumbnail (each video averaged around 5 reasonable frames). When judging reasonableness, we took into account the options available for the model, so a frame that for one video may have been deemed reasonable could, if it had been associated with another video, been deemed unreasonable. We also identified a top choice for each video. To gauge the success of the model, we are therefore able to use two metrics:

1. The percentage of the frame selector's choices that fell within our "reasonable" frames
2. The percentage of the frame selector's choices that matched our top choices identically

Though somewhat subjective, these metrics were essential for giving us more than simply qualitative results on our downstream task.

4. Experiments

4.1. Convolutional Classifier

Our best basic model (i.e., trained from scratch and using only the architecture explained above) is based on the AlexNet [18] architecture with the following modifications:

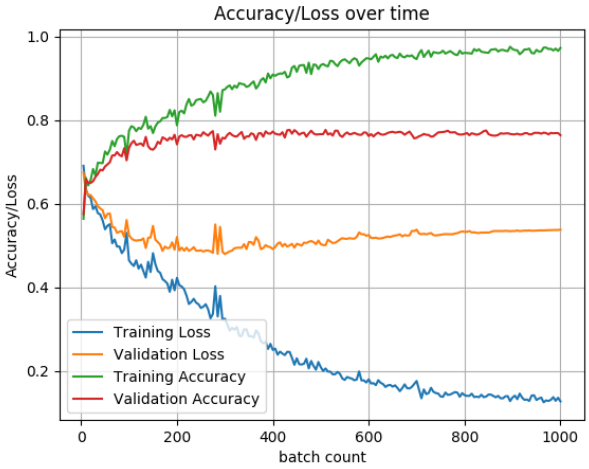
1. We removed the batch normalization layers because they did not help learning.
2. We decreased the filter size in the first convolutional layer from 11×11 to 5×5 because our images have about half as many pixels as ImageNet [11], which AlexNet was trained on.
3. We reduced the size of the dense layers from 4096 units to 1000 because we are only performing binary classification.

A visualization of this model is included in section 3.3. The hyperparameters we tuned for this model were the learning rate for the Adam Optimizer, the Dropout drop percentage, the L2 regularization and the learning rate decay. We used step-wise exponential learning rate decay every 50 iterations.

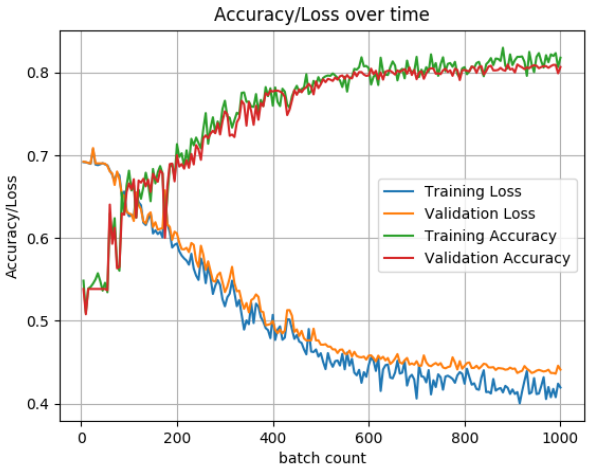
Because it takes a long time to train the full model, we used a model with half as many filters in each convolutional layer to perform hyperparameter tuning, making the assumption that the best set of hyperparameters on the smaller model will also perform well on the full model. Below are the results of our tests on the half-sized model. The final test was on the full-sized model, in which we used a slower LR Decay and more iterations:

Validation Accuracy	Learning Rate (LR)	LR Decay	Reg	Drop %
0.76	1e-4	0	0	0
0.76	1e-4	0	1e-2	0
0.5594	1e-4	0	1e-1	0
0.762366	1e-4	0	1e-2	0.2
0.763386	1e-4	0	1e-2	0.4
0.790413	1e-3	0.631	1e-2	0.4
0.809	1e-3	0.79	1e-2	0.4

In order to visualize the effects of our regularization techniques, here is the training graph for the first test in the chart:



and here is the training graph for the last one:



Since the model was hardly overfitting on the final test, we did not feel the need to increase regularization.

4.2. Evaluation of the Classifier

As mentioned above (see 3.3), because our data is noisy and number of views is not a perfect proxy for thumbnail quality, we developed a human benchmark for the classification task. Accordingly, our best model's accuracy of 80.9% is almost equal to a human level of performance (i.e., 81.6% accuracy) on the classification task.

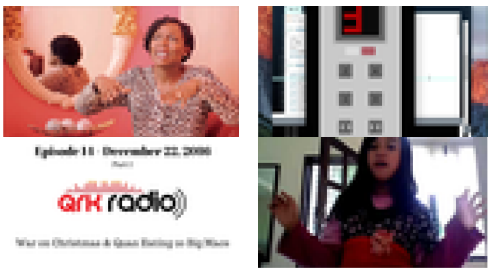
To better evaluate the model, we compared the number of false positives and false negatives. They are roughly comparable, with a 7.7% false positive rate and a 10.7% false negative rate (see confusion matrix below). The fact that the model is slightly more likely to make false negative errors is somewhat surprising, however, given that we expected that custom-made, highly-viewed thumbnails are less likely to appear poor quality than randomly-selected, rarely-viewed thumbnail are to to appear high quality. This does not appear to be the case (see examples below).

Confusion Matrix

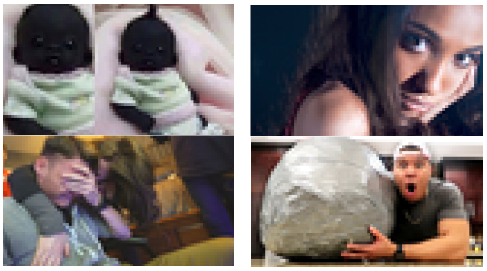
	Predicted Good	Predicted Bad
True Good	688	217
True Bad	156	900

Examining the false negatives and false positives reinforced our view that the model's mistakes to a large extent simply reflect the noise in the data. Many "mistakes"–probably at least half–come from images that we would also misclassify. And even for the instances in which the model misclassified a clearly good or bad example, odd lighting patterns often help explain the error (see below).

False Positives



False Negatives



Images on the right are mistakes we find surprising; those on the left are mistakes that seem reasonable.

We also tested whether the model's accuracy varies based on the category of the video. Overall, accuracy on the top categories is fairly consistent, suggesting that our model is not leveraging the differences in the category distributions of the good and bad examples to make predictions.

Accuracies for the 5 most common categories:

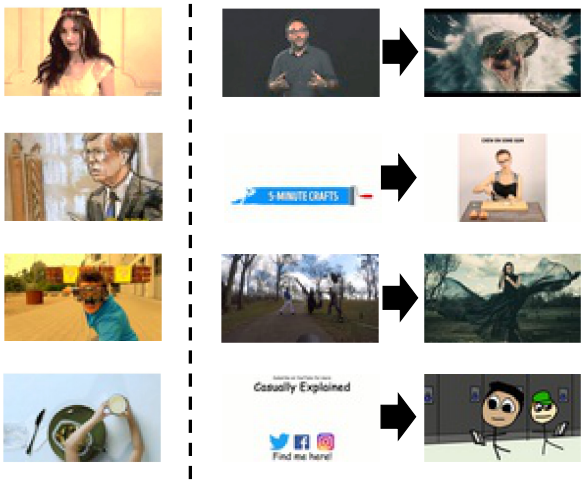
Category	Accuracy
Music	77.0%
Entertainment	75.1%
Howto & Style	75.5%
People & Blogs	85.2%
News & Politics	87.1%

To gain a better sense for the types of mistakes the model makes, we calculated saliency maps for validation data. For the most part, these maps were fairly uninformative (see below, left), which is reasonable given that image quality is a diffuse property that is unlikely to depend strongly on any one region or component. We did, however, notice that the model focused heavily on image logos (see below, right). We think that the because good thumbnails often include branding (see the Beyonce thumbnail in 3.1) the model learns to predict image quality based on the presence of a logo. To address this issue, we experimented with data augmentation (see 4.4).

Saliency Maps: the example on the left illustrates the diffuse saliency maps that commonly appear; the example on the right illustrates the model's tendency to focus almost exclusively on a logo if one is present.

4.3. Frame Selection

When run with our full variant of AlexNet, the frame selection pipeline was able to achieve 83.9% reasonableness and 23.5% agreement with our top choice (metrics as described in 3.3). Below are shown are few examples of successful and unsuccessful choices. One theme that emerges is the tendency of the model to select frames with text or logos, which it might see as more similar to the custom thumbnails of the popular channels. Human figures were also often chosen over arguably better thumbnails, as in the base of the first row on the right below. Mostly, however, our model benefited from this tendency, as evidenced by many of the successful examples shown on the left.



Left: Successful choices. Center: Sub-par choices. Right: Our preferred replacement for the sub-par choice.

4.4. Augmentation

We experimented with data augmentation for 3 reasons:

1. To expand the effective size of our dataset;
2. To address a unique instance in which our model selected an almost uniformly black frame; and
3. To address the model's tendency to evaluate images based on the presence of a logo.

Accordingly, we developed 3 methods for augmenting images:

1. Flip thumbnails horizontally
2. Jitter thumbnail pixel values with a small amount of noise
3. Insert a logo into the one of the corners or the center of the thumbnail

To select the logos, we hand-picked 100 channel thumbnails of popular YouTube channels (i.e., the image associated with the channel), selecting ones with simple geometric shapes or small bits of text to match the kinds of logos that appear on thumbnails [4]. To augment the data, for each image in the train set, we created each augmented version of the image independently with probability $= 0.4$. This roughly doubled the size of our dataset. As a final method of augmenting the dataset, we inserted "images" with random pixel values and with uniform pixel values into the dataset (labeled as bad) so that these two groups comprised 2.5% of the dataset.

Images augmented with logos in the one of the corners or the center

We tested augmentation with and without the addition of logos. Augmenting with logos reduced classification accuracy to 80.57%, while augmenting without logos increased classification accuracy to 81.43%. This indicates that adding the logos hurt the model's ability to differentiate good and bad images, an encouraging sign because it suggests that the model was forced to evaluate image quality using features other than branding. However, on the frame selection task we saw no overall improvement over the original model. (For both models, reasonableness score fell by a few % while top choice agreement rose by a few %.)

We noticed differences in the kinds of frames selected by the two augmented models. The model with logos added displays a tendency to select frames with large amounts of text where the augmented model without logo picks (good) action shots. Conversely, the non-logo model occasionally picks frames with logo-like graphics where the model with logos chose better frames. So, it seems that while including the logos allows the model to correctly ignore logo-like images when scoring frames, doing so generally degrades the model's ability to recognize high-quality images, perhaps because superimposing the logos obscures important parts of the thumbnails.

Left: Frames selected by model augmented without logos; Right: Frames selected by model augmented with logos. Each seems to make mistakes under different circumstances.

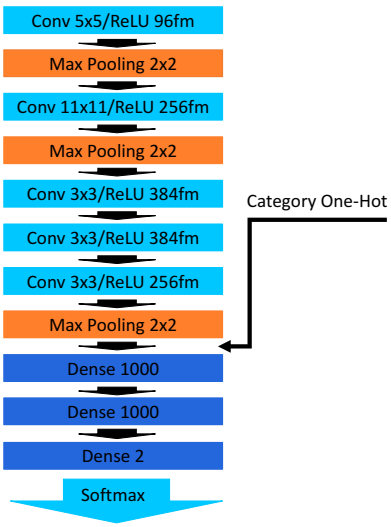
4.5. Pretrained Model: VGGNet

We also downloaded pretrained weights from the 16 layer VGGNet [23] (TensorFlow code adapted from [3]). With only retraining the final layer, we achieved 86.5% accuracy on the validation set for the classification task. However, we did not notice significant differences on our small frame selection task with this model. A larger frame selection dataset would be required to determine if higher classification accuracy always corresponds to better selection results.

It is also worth noting that we needed to stretch our images from 80×45 to 224×224 so that they would be the right size for VGGNet. This mismatch between the original image sizes and the input size to the VGGNet may have cost us some accuracy.

4.6. Categories

As a way of incorporating semantic information into our model, we created another iteration of our AlexNet variant that took a video's category as input to the dense layer. There are 44 Youtube categories (e.g. Music, Comedy, Horror) so we represented each category as a 44x1 one-hot vector. These vectors were incorporated into the model by concatenating them with the flattened output of the last convolutional layer as shown the diagram below. We then retrained the dense layers of the model.



The AlexNet variant with category information incorporated.

The results of adding category information to the model was satisfactory in the upstream classification task. With categories, the thumbnail classifier was able to achieve an accuracy of 86.23% on the validation set. Again, however, this accuracy did not translate to any significant increase

in the "reasonableness" of the model in the frame selection task. It did, however, change the qualitative behavior of the model in a few notable cases. Take the two cases shown below. On the right are the old selections of the model which seem biased towards text and human figures, while on the left are the selections of the model when passed category information. The new selections seem far more apt and exciting, likely because the model now has a sense that the fantasy movie and war documentary should be paired with thumbnails that emphasize action.

Right: Old thumbnail selections. Left: New thumbnail selections.

Because this model had the highest validation accuracy of any we implemented, we used it to evaluate accuracy on the test dataset [3]. Our accuracy on the test set was 78.06%. While this is significantly lower than our best validation accuracy, the difference seems reasonable given the small size of the test dataset (only 960 examples).

5. Conclusions and Future Work

5.1. Conclusions

Our model shows better than human levels of accuracy on the classification task on the validation set, and only slightly below human levels of accuracy on the test set. It may not be possible to achieve much greater accuracy since there are plenty of good videos with bad thumbnails and bad videos with good thumbnails.

One limitation is that our model is fitting to certain features common in the thumbnails of popular videos such as having text in the image. Text however is not indicative of a good thumbnail unless it says something that will draw the viewer into the video. Our logo-based data augmentation was able to reduce the model's preference for logos in images, so it may be possible to mitigate its preference for text by including text-based augmentation.

Based on our metrics, success in the classification task results in success on the frame selection task. However, comparing the model's choices to our human judgments makes the questionable assumption that we are capable of correctly selecting thumbnails. A larger dataset on the downstream task will be required to determine to what extent success on the two tasks are correlated with one another.

Including categories is a promising direction for this kind of project. For now we are limited by the size of our dataset and the fact that our good and bad datasets have different category distributions. We do, however, have some qualitative results indicating that the model trained with category information will make better selections once it knows the category of a video.

5.2. Future Work

There are several ways in which we could expand on our work:

1. With more time and computational resources, we could increase the resolution of the thumbnails and select the best frames from a set of more than 10 images per video. (We glanced at results using 20 and 30 frames per video; the model seemed to preform similarly well.)

2. A direct continuation of our work would be to train our classifier using additional model architectures, such as Resent and GoogleNet. Given our already high accuracy relative to human performance, however, it seems unlikely that these new models would preform significantly better, especially on the downstream, frame selection task.

3. Another expansion of our project would involve gathering much more human data for the frame selection task, as a means of better assessing the strengths of the various models we implemented or even (with enough data) to condense our pipeline so that we are training on our ability to predict the highest ranked frames.

4. Incorporating video titles and descriptions could allow our model to better select frames that are relevant to the main subject of a video. Incorporating these features, however, would require overcoming a technical and theoretical hurdle. On the technical side, many YouTube videos' titles and descriptions include characters from other languages and include many proper nouns, complication the training process (a character based model would likely be essential to encode the texts). And on the theoretical side, including titles in the model raises a risk the algorithm learns to make classification predictions without taking image quality into account.

Note: Honor Code

All authors are members of CS 231N. This project has not been submitted to a conference or journal. It has also

[3] We got our VGG implementation working only after running this test.

not been used as part of a dual-class project report. All code is our own, except for the VGG architecture, which was adapted from code provided by Olivier Moindrot [3].

References

[1] randomyoutube.net generously provided us with video IDs for random YouTube videos scraped using their pseudorandom algorithm. https://randomyoutube.net/.

[2] YouTube Creator Academy. Lesson: Make clickable thumbnails. https://creatoracademy.youtube.com/page/lesson/thumbnails#yt-creators-strategies-5.

[3] Tensorflow code for vgg adapted from. https://gist.github.com/omoindrot/dedc857cdc0e680dfb1be99762990c9c.

[4] Social Blade. Top 25 youtube users by subscribers. https://socialblade.com/youtube/.

[5] Youtube developer api. https://developers.google.com/youtube/.

[6] youtube-dl. https://rg3.github.io/youtube-dl/.

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[8] S. Bhattacharya, R. Sukthankar, and M. Shah. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 271–280. ACM, 2010.

[9] J. Choi and C. Kim. A framework for automatic static and dynamic video thumbnail extraction. *Multimedia Tools and Applications*, 75(23):15975–15991, 2016.

[10] S. Dallwig, N. Fahrer, and C. Schlier. The combination of complex scaling and the lanczos algorithm. *Chemical Physics Letters*, 191(1-2):69–76, 1992.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[12] R. A. Dunne and N. A. Campbell. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne, 181*, volume 185, 1997.

[13] Y. Gao, T. Zhang, and J. Xiao. Thematic video thumbnail selection. *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 4333–4336, 2009.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[15] J. Jiang and X.-P. Zhang. A novel video thumbnail extraction method using spatiotemporal vector quantization. In *Proceedings of the 3rd international workshop on Automated information extraction in media production*, pages 9–14. ACM, 2010.

[16] E. Jones, T. Oliphant, and P. Peterson. {SciPy}: open source scientific tools for {Python}. 2014.

[17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] H.-C. Lian, X.-Q. Li, and B. Song. Automatic video thumbnail selection. In *Multimedia Technology (ICMT), 2011 International Conference on*, pages 242–245. IEEE, 2011.

[20] C. Liu, Q. Huang, and S. Jiang. Query sensitive dynamic web video thumbnail generation. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2449–2452. IEEE, 2011.

[21] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3707–3715, 2015.

[22] A. Y. Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Y. Song, M. Redi, J. Vallmitjana, and A. Jaimes. To click or not to click: Automatic selection of beautiful thumbnails from videos. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 659–668, New York, NY, USA, 2016. ACM.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[26] M.-h. T. Weilong Yang and T. Izo. Video thumbnail selection based on deep learning. *Technical Disclosure Commons*.

[27] Y. Z. C. C. Wu Liu, Tao Mei and J. Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. *Conference on Computer Vision and Pattern Recognition*, 2015.

[28] W. Yang and M. hsuan Tsai. Improving youtube video thumbnails with deep neural nets. *Google Research Blog*, 2015.

[29] R. Zhang, S. Tang, W. Liu, and J. Li. Multimodal tag localization based on deep learning. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, page 50. ACM, 2015.

[30] R. Zhang, S. Tang, W. Liu, Y. Zhang, and J. Li. Multi-modal tag localization for mobile video search. *Multimedia Systems*, pages 1–12, 2016.

[31] W. Zhang, C. Liu, Q. Huang, S. Jiang, and W. Gao. A novel framework for web video thumbnail generation. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*, pages 343–346. IEEE, 2012.

[32] W. Zhang, C. Liu, Z. Wang, G. Li, Q. Huang, and W. Gao. Web video thumbnail recommendation with content-aware analysis and query-sensitive matching. *Multimedia Tools Appl.*, 73(1):547–571, Nov. 2014.